

AMENDMENT

IN THE CLAIMS

Please amend claims 1-30 as follows:

Claims 1-15 (canceled)

Claim 16 (previously presented): A database reorganization system, comprising data records for holding data entries, each data record contain a primary key; primary blocks for storing data records in the order of the primary keys thereof; overflow blocks linked to the primary blocks; a current location table and a new location table for containing in contiguous regions entries describing the addresses of the primary blocks;

a current location table reorganization pointer ~~for current location table~~ that indicates through which entry in the current location tables reorganization has completed;

~~a final pointer for the current location table;~~ and

a new location table reorganization pointer ~~for the new location table~~ that indicates through which entry in the new location table reorganization has completed; and

a current location table final pointer that indicates the final position used by that location table.

Claim 17 (previously presented): The database reorganization system of claim

16, wherein

the database recognition system is configured to sequentially write entries in the current location table to the new location table and, where any overflow block is present, to delink said overflow blocks, creating new entries corresponding to the primary blocks and adding the new entries to the new location table.

Claim 18 (currently amended): ~~The~~ A method of reorganizing the database reorganization system of claim 16, ~~further~~ comprising steps of:

~~a first means for~~, upon receipt of a database reorganization command, creating a new location table in addition to the current location table; ~~and~~

~~a second means for~~ sequentially writing entries in the current location table to the new location table; ~~and~~[[,]]

when an overflow block~~s~~ linked to a primary block is detected, delinking that overflow blocks, adding new entries to the new location table, and rendering the overflow blocks as new primary blocks.

Claim 19 (currently amended): ~~The~~ A method of reorganizing the database reorganization system of claim 16, ~~further~~ comprising steps of:

~~means for~~ shifting ~~fore and aft~~ before and after records in primary blocks and eliminating fragmentation when a storage rate in primary blocks falls outside a range of predetermined values; and

~~means for~~ sequentially writing entries in the current location table to the new location table.

Claim 20 (canceled)

Claim 21 (previously presented): ~~The~~ A method of reorganizing the database reorganization system of claim 16, ~~further comprising steps of:~~

~~a comparative means for, when retrieving a record by with the primary key during reorganization, comparing the value of~~ evaluating whether the target primary key with the value ~~of~~ is greater than or less than the primary key of the record contained in the primary block and the overflow blocks ~~of the entry indicated by at least one of that~~ the reorganization pointers is pointing to; and

~~a retrieval means for using the current location table to retrieve the target record when the value of the target primary key is found by the comparative means to be greater than or equal to the value of the primary key of the record stored in the blocks indicated by at least one of said reorganization pointers and for using the new location table to retrieve the target record when it is found to be less than the value of the primary key~~

when the target primary key is evaluated to be greater than or equal to the primary key of the record stored in the block that the reorganization pointer is pointing to, using the current location table to retrieve the target record; and

when the target primary key is evaluated to be less than that primary key, using the new location table to retrieve the target record.

Claim 22 (previously presented): A database reorganization system, comprising:
data records for holding data containing primary keys and alternate keys;

alternate-key entries that hold data entries, each alternate-key entry comprises an alternate key and a primary key;

alternate-key blocks for containing the alternate-key entries;

alternate-key overflow blocks linked to the alternate-key blocks;

a current alternate-key location table and new alternate-key location tables for containing alternate-key location table entries in contiguous regions;

a current alternate-key location table reorganization pointer ~~for current alternate-key location table which~~ that indicates a progress of ~~recognition~~ reorganization of the alternate-key location table and alternate-key blocks for the current alternate-key location tables;

~~a final pointer which indicates a final point of the most currently used entry of the alternate-key location table for the current alternate-key location tables; and~~

~~a reorganization pointer for the new alternate-key location table~~

a new alternate-key location table reorganization pointer that indicates a progress of reorganization of the alternate-key location table and alternate-key blocks for the new alternate-key location table; and

an alternate-key final pointer that is provided to the current alternate-key location table to indicate the final position used by said alternate-key location table.

Claim 23 (previously presented) ~~The~~ A method of reorganizing the database reorganization system of claim 22, ~~further~~ comprising steps of:

~~means for~~ sequentially writing entries in current alternate-key location tables to a new alternate-key location table and, where ~~an~~ alternate-key overflow blocks exists,

delinking the alternate-key overflow blocks, creating new alternate-key location table entries corresponding to the alternate-key blocks, and

adding new alternate-key location table entries to a new alternate-key location table.

Claim 24 (previously presented): ~~The~~ A method of reorganizing the database reorganization system of claim 22, ~~further~~ comprising steps of:

~~a first means for~~, upon receipt of a database reorganization command, creating a new alternate-key location table in addition to the current alternate-key location tables; and

~~a second means for~~ sequentially writing entries in the current alternate-key location table to the new alternate-key location table; and,

when alternate-key overflow block linked to alternate-key block is detected, delinking that alternate-key overflow block, adding new alternate-key location table entries to new alternate-key location table and rendering these as new alternate-key blocks.

Claim 25 (previously presented): ~~The~~ A method of reorganizing the database reorganization system of claim 22, ~~further~~ comprising steps of:

~~means for~~ shifting ~~fore and aft~~ before and after records in the alternate-key blocks and eliminating fragmentation when the storage rate in the alternate-key blocks falls outside a range of the specified values; and

~~means for~~ sequentially writing entries in the current alternate-key location table to new alternate-key location table.

Claim 26 (previously presented): ~~The~~ A method of reorganizing the database reorganization systems of claims 22, 23 or 24, further comprising steps of:

~~a comparative means for, when retrieving a record by~~ with the alternate key during reorganization, ~~comparing the value of~~ evaluating whether the target alternate key ~~with the value of~~ is greater or less than the alternate key of the ~~record~~ entry contained in the alternate-key block ~~of the entry indicated by at least one of~~ that said reorganization pointer is pointing to; and

~~a retrieval means for using the current alternate-key location table to retrieve the target~~ record entry when ~~the value of the target alternate key is found~~ evaluated by the comparative means to be greater than or qual to ~~the value of the alternate key of the~~ record entry stored in the alternate-key blocks ~~indicated by at least one of~~ that the reorganization pointer ~~and for~~ is pointing to; and

using the new alternate-key location table to retrieve the target ~~record~~ entry when it ~~is found~~ the target alternate key is evaluated to be less than the ~~value of that~~ alternate key.

Claims 27-30 (canceled)